Deception Considered Harmful

John J. Grefenstette
Navy Center for Applied Research in Artificial Intelligence
Code 5514
Naval Research Laboratory
Washington, DC 20375-5000
E-mail: GREF@AIC.NRL.NAVY.MIL

Abstract

A central problem in the theory of genetic algorithms is the characterization of problems that are difficult for GAs to optimize. Many attempts to characterize such problems focus on the notion of *deception*, defined in terms of the static average fitness of competing schemas. This note argues this popular approach appears unlikely to yield a predictive theory for genetic algorithms. Instead, the characterization of hard problems must take into account the basic features of genetic algorithms, especially their dynamic, biased sampling strategy.

Keywords: Deception, building block hypothesis

1 INTRODUCTION

Since Holland's early work on the analysis of genetic algorithms (GAs), the usual approach has been to focus on the allocation of search effort to subspaces described by schemas representing hyperplanes of the search space. The Schema Theorem (Holland, 1975) provides a description for the growth rate of schemas that depends on the observed relative fitness of the schemas represented in the population. Bethke (1981) initiated work on the formal characterization of problems that might be difficult for GAs to solve, and presented an analysis of problems in terms of the static analysis of schema fitness.

Goldberg (1987) introduced the notion of "deception" in GAs, and defined a Minimal Deceptive Problem (MDP). Goldberg's experiments showed that GAs could usually optimize the MDP. Nonetheless, "deception" is now widely regarded as a necessary feature in problems that are difficult for GAs (Das and Whitley, 1991; Homaifar et al, 1991). Goldberg and his colleagues (Goldberg, Deb and Korb, 1991) have defined messy GAs (mGAs) specifically to handle deceptive problems, and consider the use of deceptive functions as test functions to be "critical to understanding the convergence of mGAs, traditional GAs, or any other similarity-based search technique". The literature on deception in GAs is growing rapidly (Battle & Vose, 1991; Goldberg 1989a, 1989b, 1989c; Goldberg, Deb and Korb, 1991; Liepins & Vose, 1991; Mason, 1991; Whitley, 1991, 1992), so this certainly a topic that deserves careful scrutiny.

In previous papers (Grefenstette and Baker, 1989; Grefenstette, 1990) we have raised some questions about this approach to the analysis of GAs, and others have begun to ask similar questions (Mitchell and Forrest, 1991). This paper will try to clarify and expand on the argument that the current definitions of deception are based on faulty assumptions about the dynamics of GAs. We only address definitions of deception that are based on the static analysis of hyperplanes. By static analysis, we mean the analysis based on the average fitness of hyperplanes, when the average is taken over the entire search space. Our fundamental point is that the dynamic behavior of genetic algorithms simply cannot be predicted on the basis of the static analysis of hyperplanes. The remainder of the paper is organized as follows: Section 2 discusses the Strong Building Block Hypothesis (SBBH) that appears to underly much of the work on static hyperplane analysis. The next two sections present counterexamples to the SBBH. Section 3 shows that some functions that are highly deceptive according to the SBBH are, in fact, very easy for GAs to optimize. Section 4 shows that some functions that have no deception and therefore should be easy, according to the SBBH, are nearly impossible for GAs to optimize. These counterexamples show that deception is neither necessary nor sufficient to make a problem difficult for GAs. More importantly, the analysis of these results highlights the shortcomings of the static analysis of hyperplane. As an aside to our main point, Section 5 shows that, even if the SBBH were true, all deceptive functions could be easily solved by simple changes to the basic GA. Some final comments are contained in Section 6.

2 THE STRONG BUILDING BLOCK HYPOTHESIS

This paper is concerned with the fundamentals of how genetic algorithms work. Of course, genetic algorithms are dynamically complex processes, and we do not yet have much of a formal grasp on their dynamics. The fundamental theorem is certainly the Schema Theorem (Holland, 1975) that describes the relationship between the expected growth in a hyperplane from one generation to the next as a function of the hyperplane's observed relative fitness (as well as other factors such as the influence of crossover and mutation). The Schema Theorem is only directly applicable to a single generational cycle, but it is natural to try to extrapolate it over many generations in order to understand the long term dynamics of the GA. A formal extension of the Schema Theorem is not yet available[1], but there have been informal arguments about how the Schema Theorem plays out over time, including Holland's analogy to a k-armed bandit problem. These informal accounts of the dynamics of GAs have become part of the folklore of the field. For example, Goldberg (1989a, p. 41) asserts the "Building Block Hypothesis": "[A] genetic algorithm seek[s] near optimal performance through the juxtaposition of short, low-order, high performance schemata, or building blocks."

We will examine a slightly more operational version of this proposition, which we call:

> The *Strong Building Block Hypothesis* (SBBH): GAs proceed by finding low order schemas with the best static average fitness in each hyperplane partition and using these to build up more complete solutions.

This hypothesis clearly underlies much of the recent published work in GA theory, especially work on deception. For example, Goldberg (1989a) introduces the Minimal Deceptive Problem as follows:

> "[W]e still need to understand better what makes a problem difficult for a simple GA. To investigate this matter further, let's construct the simplest problem that should cause a GA to diverge from the global optimum ... To do this, we want to violate the building block hypothesis in the extreme." (p. 46)

The SBBH implies that functions for which the schemas associated with the optimum have higher static average fitness than the competing schemas in their partitions ought to be easy for GAs. For example, suppose the global optimum is 000 ... 0, and that

$$
\begin{aligned}
&f(0\#...\#) > f(1\#...\#)\\
&f(00\#...\#) > f(01\#...\#)\\
&f(00\#...\#) > f(10\#...\#)\\
&f(00\#...\#) > f(11\#...\#)
\end{aligned}
$$

and so on for every hyperplane partition of the search space.[2] According to the SBBH, a GA should find f simple to optimize. In fact, such functions are commonly called "GA-

[1] A previous paper suggested some ideas in this direction (Grefenstette, 1990).

[2] In this paper, $f(S)$ refers to the static average fitness for schema S, that is, the mean fitness value of every point described by that schema. This *static* average is independent of whatever points happen to be in the population at any time.

easy" (Wilson, 1991).[3]

Conversely, the SBBH implies that functions for which which the schemas associated with the optimum have lower static average fitness than the competing schemas in their partitions ought to be difficult for GAs. For example, suppose the global optimum is 000 ... 0, and

$$f(0\#...\#) < f(1\#...\#)$$
$$f(00\#...\#) < f(01\#...\#)$$
$$f(00\#...\#) < f(10\#...\#)$$
$$f(00\#...\#) < f(11\#...\#)$$

and so on. Then this function would be called "deceptive" (Goldberg, 1987).[4]

The SBBH is an appealing intuitive explanation for how GAs work, but it does not follow from the Schema Theorem. The Schema Theorem describes the expected growth of a hyperplane for a single generation based on its *observed* average fitness, that is, based on the average fitness of current samples of the hyperplane in the population. Over a period of generations, the observed average fitness of a hyperplane does not necessarily reflect the static average fitness of the hyperplane. The SBBH arises when we ignore the crucial distinction between observed average fitness and static average fitness. Consequently, while the Schema Theorem certainly applies to GAs, the SBBH does not.

There are at least two reasons why the SBBH fails to hold:

1. Biased sampling due to previous convergence.
2. Large variance within schemas.

The first factor is the most ubiquitous, since it arises even with a large population and even if the variance within individual schemas are not large (but greater than 0). Once the population begins to converge, even a little, it is no longer possible to estimate the static average fitness of schemas using the information present in the current population. The GA can only estimate the *conditional* average fitnesses, conditioned by the converged alleles. That is, after the very first generation, the population represents a heavily biased sample of all schemas. Admittedly, it is very hard to analyze the conditional estimates of fitnesses, and so it may be tempting to limit our attention to what happens at generation 0, but this is a pretty uninteresting period in the life of a GA.

---

3 We are not asserting that Wilson subscribes to the SBBH.

4 Goldberg's experiments show that deceptive problems are not necessarily hard. However, he calls his own results "surprising" (Goldberg, 1989a, p. 46, p. 51), presumably because they violate the SBBH.

In the next two sections, the reasons listed above are used to show why the notion of deception is of so little use in predicting how difficult a function may or may not be for a GA to optimize.

3 BIASED SAMPLING DUE TO PREVIOUS CONVERGENCE

The primary reason that analysis based on static average fitness of schema is a dead end is that, except possibly for the very first generation, the population contains only a biased sample of representatives from each schema. This is a normal feature of all GAs, but it can yield results that are exactly the opposite of what one might expect from the SBBH.

Using this observation, it is a simple exercise to define problems that are highly deceptive in the sense implied by the SBBH, but are actually easy for GAs to optimize. Here is one example: Consider a 10-bit space representing the interval [0.0, 1.0] in binary encoding. That is, 0000000000 represents 0.0 and 1111111111 represents 1.0. We want to maximize f, defined by:

$f(x) = x^2$, except for the following special cases:

$f(0111111111) = 1.01$
$f(0011111111) = 1.02$
$f(0001111111) = 1.03$
$f(0000111111) = 1.04$
$f(0000011111) = 1.05$
$f(0000001111) = 1.06$
$f(0000000111) = 1.07$
$f(0000000011) = 1.08$
$f(0000000001) = 1.09$
$f(0000000000) = 1.10$

According to Whitley (1991), this function is "Order 9 Deceptive"; that is, an enumeration of all schema competitions shows that except for 0000000000 and 000000000#, every schema representing the optimum has a static average fitness less than the competing schema representing the suboptimal 1111111111. Despite this high level of deception, a standard GA (GENESIS) with population size 100 and the default parameter settings finds the optimum after a few thousand trials.[5] What happens in practice is that the GA rapidly converges toward the suboptimal 1111111111. Once the

[5] One change to GENESIS was necessary -- the code that normally causes an abort after nearly converging was disabled.

population has nearly converged, the special cases each successfully propagates, one at a time.

Although this example is deliberately defined to be as simple as possible to allow us to completely understand the dynamic of the genetic search, it is hoped that the reader will easily see how the same general phenomena can happen in naturally occurring problems. In any problem, as the search proceeds, the convergence in the population radically alters the competition between competing hyperplanes, so that the static average fitnesses become completely irrelevant. In the example, once the population has largely converged to 1111111111, then the competition between the schema 1# ... # and 0#...# in fact becomes a competition between 1111111111 (with fitness 1.0) and 0111111111 (with fitness 1.01), and the latter prevails. Once the population converges largely to 0111111111, the competition between #0######## and #1######## becomes a competition between between 0011111111 (with fitness 1.01) and 0111111111 (with fitness 1.02).

Although this artificial example is an extreme case, it illustrates a perfectly normal phenomena in GAs: the normal convergence of the population produces outcomes that are at variance with the SBBH, because the samples from the competing hyperplanes are highly biased. The example also suffices to show that some highly deceptive problems are easy for GAs.[6]

4 LARGE VARIANCE WITHIN SCHEMA

The second shortcoming of the static analysis of schema fitness is that it usually ignores the effects of variance of fitness within schemas. With a limited population size and large variance within the schemas, even the sampling in the initial, random population will produce errors in the estimate of each schema's static average fitness. This schema variance can also lead to results that are at variance with the SBBH.

As an example, we can define a class of problems that are "easy" in the sense implied by the SBBH -- they have no deception -- but are in fact hard for GAs to optimize. Consider a L-bit space representing the interval [0.0, 1.0] in binary encoding. Let f be defined:

$$f(x) = x^2 \quad \text{if } x > 0,$$

[6] Some defenders of deception have argued that this example is flawed because it depends on mutation to keep the competition alive when the selective pressure drives the population toward complete convergence. This is a curious argument since it seems to imply that deception is a problem only when there is no mutation in the GA. It should be noted that the GA that solves this example uses a very common low mutation rate (0.001).

$$f(0) = 2^{(L+1)}$$

For any schema S such that the optimum is in S (that is, all the defined positions of S have value 0), $f(S) > 2$, since the sum of the fitness of the points in S is at least $2^{(L+1)}$ and there are at most 2^L points in the hyperplane. For any schema S such that the optimum is not S, $f(S) \leq 1$. So in any schema partition, the schema containing the optimum has the highest static average fitness. That is, there is no deception at any level in the function. Such functions are often called "GA-easy" (Liepins and Vose, 1991; Wilson, 1991).

Suppose we run a standard GA on f with a population of size 100. If the optimum is not in the initial population, it will probably never be found. (Of course, it might be created by a lucky crossover or a very lucky multiple mutation.) Why is this function hard for GAs? Because the schemas associated with the optimum have extremely high variance, so the observed average fitness for the hyperplanes never reflects their static average fitnesses, not even in the initial random population. Of course, this is a "needle-in-a-haystack" function, so we don't expect the GA to solve it on a regular basis. But it does satisfy the commonly used definition of "GA-easy" that follows from the SBBH, so this example provides a counterexample to the claim that only deceptive problems are challenging for GAs (Das and Whitley, 1991).

In some cases with less extreme variance in the schemas, we might be able to reduce the variance associated with the observed fitness of a schema by increasing the population size. Based on this observation, it is tempting to think that a GA with a large population size might actually conform to the SBBH. This is not that case. Even in the unlikely event that we would run a GA with population size of 2^L or more, the GA would still compute a biased estimate for the fitness of schema after the initial generation. See (Grefenstette & Baker, 1989) for an example. Having a large population does not in any way ensure that one will have "enough samples to provide reliable estimates of hyperplane fitness" (Spears & De Jong, 1991), since the samples will *not* be chosen from a uniform distribution which each schema, but instead will be chosen is a highly biased way, emphasizing the elements of each schema that are more highly fit. Large populations cannot save the SBBH.

There has been some recent work that addresses the (static) fitness variance within schemas (Goldberg and Rudnick, 1988; Rudnick and Goldberg, 1991). This is certainly a step in the right direction, but it is unlikely that an analysis of the static fitness variance will be any more helpful than an analysis of the static fitness averages. As the search proceeds, the *observed* variance associated with hyperplanes in the population is unlikely to have correlation with the *static* variance.

5 AUGMENTED GAs SOLVE DECEPTIVE PROBLEMS

As the two examples above show, the relationship between deception and GA-hardness seems pretty tenuous. Nevertheless, there is a growing literature concerning how to make GAs more effective on deceptive problems. For example, Liepins and Vose (1991) specify representation transformations that render deceptive problems "fully easy". Goldberg at. al (1991) define messy GAs in order to deal with problems with bounded deception. In this section, we show that, even if the SBBH were true, slight changes to the basic GA would be sufficient to solve most deceptive problems that have been studied.

Much of the work on deception involves functions for which the bit-wise complement of the global optimum is the deceptive attractor (Liepins and Vose, 1991; Whitley, 1991). In fact, arguments have been made that all deceptive problems have this feature (Whitley, 1992). For the purpose of this section, let us accept this argument and let us suppose that a GA would actually perform according to the SBBH on these fully deceptive problems. That is, we suppose the GA really does converge to the complement of the global optimum. Leaving aside for the moment the interesting question of whether this behavior should actually count as a failure of the GA (Mitchell & Forrest, 1991), we can easily describe an augmented GA that finds the global optimum in all such problems.

The augmented GA shown in Figure 1. The lines marked with (*) represent the only changes to a standard GA. In the augmented version, we maintain two separate populations of size N, called P and Q. During each generation, we update P according to the original GA, and set the members of Q to the bitwise complement of the corresponding elements of P.

Consider any problem for which the original GA (i.e., the one without the (*) lines) finds an acceptable solution in time t, using a population of size N. Then the augmented algorithm finds a solution that is at least as good as the original GA, in at most twice the amount of time (assuming that the evaluation time dominates the other operations in the algorithms). In addition, the augmented algorithm solves any fully deceptive problem that has the property that the global optimum is the binary complement of the deceptive attractor, since as soon as the population P produces a copy of the deceptive attractor, the population Q produces a copy of the global optimum. Thus the augmented GA can produce a final answer that is never worse than the one produced by the original GA, and it eliminates the problem of deception, all at a cost of only doubling the computational time.[7] A reasonable conclusion of this discussion is that deception is simply a non-

[7] There are many possible variations on this theme. For example, it may be desirable to allow recombination across populations *P* and *Q* on "partially deceptive" problems. Exploration of these variations

```
procedure Augmented GA
begin
        t = 0;
        initialize P(t);
(*)     Q(t) = complement(P(t);
        evaluate structures in P(t);
        while termination condition not satisfied do
        begin
                t = t + 1;
                select P(t) from P(t-1);
                alter structures in P(t);
(*)             Q(t) = complement(P(t));
                evaluate structures in P(t);
(*)             evaluate structures in Q(t);
                output best structure in P(t) ∪ Q(t);
        end
end.
```

Figure 1: A Genetic Algorithm.

problem for GAs. It can easily be handled by a low-cost alteration of the basic GA.

In practice, it would be unwise to actually implement this "solution", unless one really accepts that the GA performs according to the SBBH. Since this is not the case, the above solution would in all probability be a mere waste of effort.

6 SUMMARY

This note criticizes the notion of deception in GAs that arises from the Strong Building Block Hypothesis. According to the SBBH, deceptive problems ought to be difficult for GAs to solve, and "GA-easy" problems ought to be easy for GAs to solve. We have identified two reasons why the SBBH is false:

1. Biased sampling due to previous convergence.

---

will be deferred until a more robust form of "deception" has been identified.

2. Large variance within schemas and limited population size.

Taking these reasons into account, it is easy to demonstrate that:

1. Some highly deceptive problems are easy for GAs to optimize.

2. Some "GA-easy" problems with no deception are nearly impossible for GAs to optimize.

The examples show that, at the very least, the term *deceptive* is poorly chosen. More importantly, the examples illustrate that it is in general impossible to predict the dynamic behaviors of GAs on the basis of the static average fitness of hyperplanes. Finally, the previous section showed that, even assuming that GAs perform according to the SBBH, deceptive problems can be solved by a simple augmented GA. Thus, deception as it is usually defined is truly a non-problem for GAs.

Some of the concerns about the notion of deception have been raised by others, in particular Mitchell and Forrest (1991). These points bear further examination in this forum, to combat the apparent trend among new researchers in GA Theory to continue to flock to the analysis of deception. It is disturbing that much of what passes for GA Theory seems to assume the SBBH as a starting point. If this continues, we will reach a point where "GA Theory" deals with algorithms (if there are any) that satisfy the SBBH. Unfortunately, the GA is not one of those algorithms.

It goes without saying that the characterization of hard problems should remain a high priority for the GA research community. However, the characterization must take into account the basic features of the GA, especially its dynamic, biased sampling strategy. Our primary point is that the effort currently being expended on the static analysis of functions should be diverted to the dynamic analysis of GAs. One might argue that the work to date on deception has been a preliminary exploration of how to analyze simple distributions of fitness, and was always intended to be replaced by dynamic analysis. It is gratifying to see some recent efforts in this direction (Bridges and Goldberg, 1991; Liepins and Vose, 1991). Nevertheless, it is important that articles on GA theory avoid the implicit assumption of the SBBH. The SBBH is such an attractive and intuitive explanation for the power of the GA that it can easily mislead newcomers to the field, as well as potential users of the technology.

REFERENCES

Battle, D. L. and M. D. Vose (1991). Isomorphisms of genetic algorithms. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Bethke, A. D. (1981). Genetic algorithms as function optimizers. Doctoral dissertation, University of Michigan.

Bridges, C.L. and D. E. Goldberg (1991). The nonuniform Walsh-schema transform. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Das, R. and D. L. Whitley (1991). The only challenging problems are deceptive. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 116-173). San Diego, CA: Morgan Kaufmann.

Goldberg, D. E. (1987). Simple genetic algorithms and the minimal deceptive problem. In L. Davis (ed.), *Genetic algorithms and simulated annealing* (pp. 74-88). London: Pitman.

Goldberg, D. E. (1989a). *Genetic algorithms in search, optimization, and machine learning.* Reading: Addison-Wesley.

Goldberg, D. E. (1989b). Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems, 3,* 129-152.

Goldberg, D. E. (1989c). Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems, 3,* 153-171.

Goldberg, D. E., K. Deb and B. Korb (1991). Don't worry, be messy. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 24-30). La Jolla, CA: Morgan Kaufmann.

Goldberg, D. E. and M. Rudnick (1988). Genetic algorithms and the variance of fitness. *Complex Systems, 2,* 265-278.

Grefenstette, J. J. and J. E. Baker (1989). How genetic algorithms work: An critical look at implicit parallelism. *Proceedings of the Third International Conference of Genetic Algorithms* (pp. 20-27). Fairfax, VA: Morgan Kaufmann.

Grefenstette, J. J. (1991). Conditions for implicit parallelism. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University Michigan Press.

Homaifar, A., X. Qi and J. Fost (1991). Analysis and design of a general GA deceptive problem. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 196-203). San Diego, CA: Morgan Kaufmann.

Liepins, G. E. and M. D. Vose (1991). Deceptiveness and genetic algorithm dynamics. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Mason, A. J (1991). Partition coefficients, static deception and deceptive problems for non-binary alphabets. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 210-214). San Diego, CA: Morgan Kaufmann.

Mitchell, M. and S. Forrest (1991). What is deception anyway? Research Note, 11 July 1991.

Rudnick, M and D. E. Goldberg (1991). Signal, noise, and genetic algorithms. IlliGAL Report No. 91004, Univ. Illinois at Urbana-Champaign. Also available as Oregon Graduate Institute Technical Report No CS/91-013.

Spears, W. M. and K. A. De Jong (1991). An analysis of multi-point crossover. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Whitley, L. D. (1991). Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Whitley, L. D. (1992). Deception, dominance and implicit parallelism in genetic search. To appear in *Annals of Mathematics and Artificial Intelligence.*

Wilson, S. W. (1991). GA-easy does not imply steepest-ascent optimizable *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 85-89). San Diego, CA: Morgan Kaufmann.